UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/836,226	04/18/2001	Kevin John Moore	169.2025	6435

5514 7590 01/04/2006

FITZPATRICK CELLA HARPER & SCINTO
30 ROCKEFELLER PLAZA
NEW YORK, NY 10112

EXAMINER	
WOODS, ERIC V	

ART UNIT	PAPER NUMBER
2672	

DATE MAILED: 01/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No. 09/836,226	Applicant(s) MOORE, KEVIN JOHN	
	Examiner Eric V. Woods	Art Unit 2672	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 08 July 2005 and 09 September 2005.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-3,5-14,16-25 and 27-33 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) 7-11,18-22 and 29-33 is/are allowed.

6)☒ Claim(s) 1-3,5-6,12-14,16-17,23-25, and 27-28 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 8 July 2005 has been entered.

Response to Arguments

Applicant's arguments, see Remarks pages 1-11, filed 8 July 2005, with respect to the rejection(s) of claim(s) 1-33 under various statutes and references have been fully considered and are persuasive.

Therefore, the rejection of claims 1-8, 18, 19, and 22-30 under 35 U.S.C. 101 as unpatentable stand withdrawn in view of applicant's amendments to make them statutory.

The rejections of claims 4, 15, and 26 under 35 USC 103(a) are withdrawn because applicant has canceled those claims.

The rejections of claims 1-3, 5-14, 16-25, and 26-33 under 35 USC 103(a) are withdrawn in view of applicant's amendments to the independent claims, which have substantially changed the scope of the claims.

As such, the rest of applicant's arguments are moot.

However, upon further consideration, a new ground(s) of rejection is made in view of various references as below.

Examiner will however analyze some of applicant's arguments, as they are germane to the new grounds of rejection applied below.

Firstly, on Remarks page 5, applicant argues that all three traversals in the Politis patent are performed before the scan line-oriented rendering described in column, lines 31-43. This is irrelevant, because the number of times the tree is traversed, the manner in which some of those traversal are performed, the **reasons** for the traversals, and the **timing** when such conversions are performed are all limitations that are not found in the instant claims. Therefore, this line of argumentation consists entirely of limitations that have been impermissibly imported from the specification, and are thusly ignored under the standards of *In re Van Geuns*, as discussed in the previous, final Office Action.

Specification

The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1-3, 5-6, 12-14, 16-17, 23-25, and 27-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Politis in view of Curtis et al (US 6,330,003 B1, eligible under 35 USC 102(e)) and Katzenberger.

Please note: claims 12-17 are treated with 1-6, as 12-17 are the apparatus implementing 1-6, claims 23-28 are referenced to 1-6 and 12-17 with the additional limitation of computer-readable media covered beforehand. As to claims 23-28, the references utilize computers to implement the method, which inherently requires that the computer have a program implementing the method.

As to claims 1, 12, and 23,

A method of rendering a directed acyclic graph into a raster pixel image having a plurality of pixel locations, the directed acyclic graph comprising one or more parent nodes and one or more leaf nodes, each parent node representing an operator and having branches to respective descendent nodes, and each leaf node representing a graphic object having object edges, said method comprising the steps of: (Politis discloses the use of expression trees (see col. 8, lines 16-40). Furthermore, the "expression trees" shown in Figures 16-18 are valid forms of trees, which are by

definition undirected acyclic graphs. Next, Politis teaches that these trees are decomposed into lower-level instructions (see col. 4, lines 3-8). Politis teaches these graphs as trees, but (see Figs. 7, 8, 16, and others) they have leaf nodes (see col. 11, lines 30-33) and parents (see col. 14, lines 25-32). The parent nodes are found to be operators (element 26, Fig. 17) as shown in Figs. 16, 17, and 18 (see col. 13, lines 33-60). The leaf or descendent nodes are shown in the above figures (elements 20, 24, etc.) and are representing graphical objects such as letters (see col. 13, lines 33-60). Therefore, all the recited portions of the preamble to claim 1 and 12 are met. Rejection of the method steps follows below.)(Curtis clearly teaches a DAG (e.g. inherently a tree) in Figures 7, 15, and 16 wherein parent nodes represent operators and leaf nodes represent graphic objects having object edges – edges are discussed below in the second clause)(Katzenberger discloses the use of directed acyclic graphs to store graphical information in Figs. 2 and 4B, particularly Fig. 4B (see col. 1, lines 9-11). Parent and child nodes are used in this invention (see col. 2, lines 53-65); child nodes are the same thing as the "leaf" nodes referenced by applicant (see col. 1, lines 55-65). Furthermore, Katzenberger teaches the use of operators for performing operations on graphical areas – Fig. 14 shows the use of various operators such as 'cno' and 'ctg' that meet the requirement of the claim, even if they primarily emphasize the data relationships. As seen in Fig. 4B, Katzenberger shows the simplest graphical elements as the leaf nodes.)

-Determining groups of one or more pixel locations, wherein the groups are bounded by the object edges; (Politis teaches that there are separate groups of pixel locations

determined (see col. 4, lines 15-32), and such selections are illustrated in Figs. 30 – 32, as regions of pixels are shown with their bounding boxes (see col. 11, lines 3-4).)(Curtis clearly shows in Figures 8-13 various combinations of groups of pixel location, where such regions are clearly bound by edges – note 8:25-45. Note also that Curtis uses **path** objects, where a path is defined as being "an outline of shape", as being used to fill the interior or a shape, and the like – 7:35-45. **Such a path – of an outline -- would clearly define an object by its edges**, since such a path would inherently consist of the edges or perimeters of an object. Further, Curtis defines a **region** as consisting of paths or geometric shapes (8:25-35), particularly paths, and in 8:35-45, a region is described by a combination of paths and/or other elements. Therefore, the system of Curtis uses objects that are bound by their edges in a manner that is compatible with Politis)(Clearly, having groups of pixels that are bounded by their edges would be an inherent thing, since the boundaries of image components are clearly set forth for each polygon, even if it is re-rendered for different devices as per the Curtis reference)

-Determining, for each group, a portion of the directed acyclic graph in accordance with activities of the operators, wherein the portion of the directed acyclic graph is that portion which passes data up the directed acyclic graph; (Politis -- the operators are all described as operating within a region limited by such a box (see col. 11, lines 3-4). The claim recites, "...wherein the said portion of the directed adjacency graph is that portion which passes data up the directed adjacency graph..." (Ellipses added). This language is with used with respect to the determining of the graph to be operated upon, which is clearly shown in Fig. 17 as graphical regions are selected and would *prima*

facie pass information up the tree or graph containing them. Also, Politis very clearly teaches the determining of active pixel areas (col. 9, lines 31-42) and the updating of it – that is, inactive or completed instructions are pruned from the list. Only the active leaf nodes would be placed into the active list for a given location during the drawing process - that is, as the rastering shifted, the active instruction pool would update. Again, the above means that the tree is traversed as the rasterizing / scanning process occurs; the instructions are generated and put into the active list (see col. 9, 12-25 and 31-52). Lastly, the generation step for the trees is described in the lines referenced above, e.g. the trees are constructed during each line scan.)(Politis further teaches in 9:30-45 the teachings that a list of active areas is maintained, such that the recited determining takes place, and where as noted above, information is passed upwards)(Curtis clearly shows that portions of the graph are selected – note Figure 7, and that such nodes are operated upon in an upwards manner (note 11:20-31 for example), which would require that data be passed up the tree, particularly for the situations of Figures 15 and 16, where when nodes are passed up the tree, each node forms an intermediate stage, and is passed up the tree – see Figures 8-13 and 10:55-11:12. Note that Curtis clearly only selects areas and passes information up that are relevant to each object in combination with the operators – e.g. the system starts at the lowest node and then operates upon the objects to form the desired intermediate region and passes that information up the tree, wherein as can be seen from the various intermediate figures (8-12) given as an example, the intermediate regions are the only

ones effected, which constitutes selecting active regions and the like based on the operators)

-Traversing, for each group, the determined portion of the directed acyclic graph, and (Curtis clearly traverses the tree – see for example Figure 8 for the tree, Figure 14 for the traversal flowchart in many ways)(Curtis traverses the tree **after** scan-line conversion, note for example (11:20-11:50, specifically 11:12-20, where it clearly states that the original data is scan-converted prior to performing combine and transform operations, where clearly this is selective, since the tree is traversed for that portion of the graph – Note 11:40-12:22, where various combinations of transforms can be used to generate the same output objects on a variable-per device basis, which clearly would constitute traversing each portion of the graph for the selected group, where Curtis then traverses the tree as set forth above)(Katzenberger discusses traversing a directed acyclic graph (see col. 18, lines 8-15). This step is executed as part of an operation on the data structure. As discussed in claim 1, the active nodes are read in for each pass and updated, therefore both the operators and the active leaf nodes are traversed and translated into instructions, thus meeting the requirements of the claim.) (Politis clearly traverses various nodes of the tree –8:29-35, 9:15-22, and the like, not in Figure 28 that an optimized tree is created)

-Generating, for each group, instructions for the traversed portion of the directed acyclic graph, wherein operator instructions are generated for those operators of the traversed portion of the directed acyclic graph having active branches and wherein leaf instructions are generated for those graphic objects which are active at the group of one

or more pixel locations: and (Politis clearly teaches, as discussed above, that a listing of active nodes and elements is kept and that the DAG is optimized such that only those are traversed at any given time, note the explanation in the earlier clauses on this particular point, and objects inherently exist at pixel locations)

-Executing, for each group, the generated instructions to render the graphics objects at the one or more pixels. (Politis clearly renders the objects for display on a screen at the pixel locations. See Fig. 15 and 4:55-56)(Curtis also renders objects on the screen, note Figure 13, which is the final output of various compositing operations, with intermediate results illustrated in previous figures and already discussed above.

Politis, as noted above, clearly teaches most of the limitations of the claims. Curtis teaches the rest. Curtis utilizes a bounding box approach (8:25-35) similar to that of Politis, thusly the modification and/or combination is proper, since the systems use similar principles of operation (9:45-10:15). Therefore, the line of argumentation that applicant presented in Remarks pages 8-9 is not available herein. Further, Curtis clearly does define regions by paths, which can be the outline of an object (e.g. its edges) – if applicant claims that the claim requires that such reference must **always** use edges, applicant is pointed to 7:47-51, where it is at least strongly suggested that geometric objects consist of paths. Further, the system of Curtis allows the system to specify graphical objects and/or bounding rectangles (see how the system of Curtis can obtain bounding rectangles in 8:25-35). This would not change the principle of operation of the Politis reference. Rather, since regions can be defined by rectangles

(7:35+) or paths, the system of Curtis would extend the system of Politis, and further allow it to use only the regions that overlap, not simply the bounding boxes, where the clipping region could then be defined as a path or a bounding box, as below. Note finally that Politis minimizes the bounding box in 12:40-65 and notes that the goal is to process the minimum necessary number of pixels to accomplish the desired task, and as such, the minimum conceivably sized bounding box will be a path as noted above.

Therefore, it would be obvious to one of ordinary skill in the art to combine the instruction trees of Politis with the graph structures of Katzenberger. The motivation would be to allow compiler-style optimization of rasterizing instructions and to enable the effective use of directed acyclic graphs as discussed by Politis (see Politis col. 10, lines 1-11 and 37-40) and also to derive the benefits of the DAG storage structures as shown by Katzenberger in those figures referenced above.

Further, Politis and Curtis are analogous art that are both directed to methods of optimizing and rendering graphic objects in various formats using DAGs – that is, tree formats. Curtis teaches additional information concerning the traversal of nodes in such a tree, and that such is done **after** scan-conversion, which means that the end results (particularly since the only nodes traversed would be those that were active and in the optimized tree of Politis anyway – see Figures 28 and 29, where the tree is optimized anyway) would be optimized only traverse nodes that were active in the required manner anyway. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Politis in light of the teachings of Curtis to do the above limitations for the reasons enumerated above. Additionally and more importantly,

the system of Politis does not take into account device-specific rendering issues (e.g. an object will be rendered in different ways with different devices), where the system of Curtis does, and thusly generates device-resolution specific regions that do not require scaling, and regions are freely transformable after combination, with many other limitations (2:30-57).

As to claims 2, 13, and 24, references Politis and Katzenberger teach all the limitations. Reference Katzenberger teaches a table for storing graphical instructions for acting on image components in Fig. 9, thus meeting the details recited in claim 2. See claim 1 for motivation and combination.

As to claims 3, 14, and 25, references Politis and Katzenberger teach all the limitations. Reference Politis teaches the use of a "Cliplist" in Fig. 24 that contains a listing of elements to be moved. This list is stored in a one-column tabular format as shown in the Figure, and an operator does perform this action (see col. 3, lines 20-40). This data is constructed as the tree is traversed and could easily be inserted into the table of Katzenberger shown in Fig. 9. See claim 1 for motivation and combination.

As to claims 5, 16, and 27, reference Politis discloses the use of expression trees (see col. 8, lines 16-40). Furthermore, the "expression trees" shown in Figures 16-18 are valid forms of trees, which are by definition undirected acyclic graphs. Next, Politis teaches that these trees are decomposed into lower-level instructions (see col. 4, lines 3-8). Politis teaches these graphs as trees, but (see Figs. 7, 8, 16, and others) they have leaf nodes (see col. 11, lines 30-33) and parents (see col. 14, lines 25-32). Examiner interprets these terms, e.g. leaf and parent nodes, to have the normal

meanings associated with them in graph theory. The parent nodes are found to be operators (element 26, Fig. 17) as shown in Figs. 16, 17, and 18 (see col. 13, lines 33-60). The leaf or descendent nodes are shown in the above figures (elements 20, 24, etc.) and are representing graphical objects such as letters (see col. 13, lines 33-60). Therefore, it would be obvious to one of ordinary skill in the art to combine the trees of Politis with the graphic forms of Katzenberger. The motivation would be to have a tree structure to optimize as per claim 1 (see claim 1 references and col. 4, lines 1-10).

As to claims 6, 17, and 28, reference Politis teaches the use of binary operators on the expression trees – see Figs. 28-29, for example. See claim 1 for combination and motivation.

Claims 1-3, 5-6, 12-14, 16-17, 23-25, and 27-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Webb et al (US 5,471,568 A) in view of Curtis et al (US 6,330,003 B1, eligible under 35 USC 102(e)) and Katzenberger.

Please note: claims 12-17 are treated with 1-6, as 12-17 are the apparatus implementing 1-6, claims 23-28 are referenced to 1-6 and 12-17 with the additional limitation of computer-readable media covered beforehand. As to claims 23-28, the references utilize computers to implement the method, which inherently requires that the computer have a program implementing the method.

As to claims 1, 12, and 23,

A method of rendering a directed acyclic graph into a raster pixel image having a plurality of pixel locations, the directed acyclic graph comprising one or more parent

nodes and one or more leaf nodes, each parent node representing an operator and having branches to respective descendent nodes, and each leaf node representing a graphic object having object edges, said method comprising the steps of: (Curtis clearly teaches a DAG (e.g. inherently a tree) in Figures 7, 15, and 16 wherein parent nodes represent operators and leaf nodes represent graphic objects having object edges – edges are discussed below in the second clause)(Katzenberger discloses the use of directed acyclic graphs to store graphical information in Figs. 2 and 4B, particularly Fig. 4B (see col. 1, lines 9-11). Parent and child nodes are used in this invention (see col. 2, lines 53-65); child nodes are the same thing as the "leaf" nodes referenced by applicant (see col. 1, lines 55-65). Furthermore, Katzenberger teaches the use of operators for performing operations on graphical areas – Fig. 14 shows the use of various operators such as 'cno' and 'ctg' that meet the requirement of the claim, even if they primarily emphasize the data relationships. As seen in Fig. 4B, Katzenberger shows the simplest graphical elements as the leaf nodes.)(Webb clearly teaches such limitations in Figures 12-13, with objects for rendering decomposed into various elements, with the operators being binary nodes and the leaf nodes being a graphical object having edges – see Abstract)

-Determining groups of one or more pixel locations, wherein the groups are bounded by the object edges; (Curtis clearly shows in Figures 8-13 various combinations of groups of pixel location, where such regions are clearly bound by edges – note 8:25-45. Note also that Curtis uses **path** objects, where a path is defined as being "an outline of shape", as being used to fill the interior or a shape, and the like – 7:35-45. **Such a path**

– **of an outline -- would clearly define an object by its edges**, since such a path would inherently consist of the edges or perimeters of an object. Further, Curtis defines a **region** as consisting of paths or geometric shapes (8:25-35), particularly paths, and in 8:35-45, a region is described by a combination of paths and/or other elements. Therefore, the system of Curtis uses objects that are bound by their edges in a manner that is compatible with Katzenberger)(Clearly, having groups of pixels that are bounded by their edges would be an inherent thing, since the boundaries of image components are clearly set forth for each polygon, even if it is re-rendered for different devices as per the Curtis reference)(Webb clearly teaches that such objects are made of edges (Abstract), Figures 12-14. 5:58-6:20 clearly delineates how each object is made of edges)

-Determining, for each group, a portion of the directed acyclic graph in accordance with activities of the operators, wherein the portion of the directed acyclic graph is that portion which passes data up the directed acyclic graph; (Curtis clearly shows that portions of the graph are selected – note Figure 7, and that such nodes are operated upon in an upwards manner (note 11:20-31 for example), which would require that data be passed up the tree, particularly for the situations of Figures 15 and 16, where when nodes are passed up the tree, each node forms an intermediate stage, and is passed up the tree – see Figures 8-13 and 10:55-11:12. Note that Curtis clearly only selects areas and passes information up that are relevant to each object in combination with the operators – e.g. the system starts at the lowest node and then operates upon the objects to form the desired intermediate region and passes that information up the tree,

wherein as can be seen from the various intermediate figures (8-12) given as an example, the intermediate regions are the only ones effected, which constitutes selecting active regions and the like based on the operators)(Webb clearly constructs these kinds of trees as in Figures 12-14, and 7:65-8:40 explains Figure 12, while 9:10-55 illustrates more about the scan conversion algorithm. 6:29-41, Note for example Figure 16 as well. Scan conversion takes place one line at a time, note Figure 18, and active edges are removed once they are processed – for example 10:65-11:35)

-Traversing, for each group, the determined portion of the directed acyclic graph, and (Curtis clearly traverses the tree – see for example Figure 8 for the tree, Figure 14 for the traversal flowchart in many ways)(Curtis traverses the tree **after** scan-line conversion, note for example (11:20-11:50, specifically 11:12-20, where it clearly states that the original data is scan-converted prior to performing combine and transform operations, where clearly this is selective, since the tree is traversed for that portion of the graph – Note 11:40-12:22, where various combinations of transforms can be used to generate the same output objects on a variable-per device basis, which clearly would constitute traversing each portion of the graph for the selected group, where Curtis then traverses the tree as set forth above)(Katzenberger discusses traversing a directed acyclic graph (see col. 18, lines 8-15). This step is executed as part of an operation on the data structure. As discussed in claim 1, the active nodes are read in for each pass and updated, therefore both the operators and the active leaf nodes are traversed and translated into instructions, thus meeting the requirements of the claim.)(Webb traverses the tree; note 12:15-22 (claim 1))

-Generating, for each group, instructions for the traversed portion of the directed acyclic graph, wherein operator instructions are generated for those operators of the traversed portion of the directed acyclic graph having active branches and wherein leaf instructions are generated for those graphic objects which are active at the group of one or more pixel locations: and (Webb clearly teaches, as discussed above, that a listing of active nodes and elements is kept and that the DAG is optimized such that only those are traversed at any given time, note the explanation in the earlier clauses on this particular point, and objects inherently exist at pixel locations)(Webb renders the active edges as discussed above and then takes them out of the list at the end of the process)

-Executing, for each group, the generated instructions to render the graphics objects at the one or more pixels. (Curtis also renders objects on the screen, note Figure 13, which is the final output of various compositing operations, with intermediate results illustrated in previous figures and already discussed above.)(Webb clearly scan-converts lines to render them on a display device – note Figures 1-3)

Webb as noted above, clearly teaches most of the limitations of the claims. Curtis teaches the rest. Curtis utilizes an edge or path based system (8:25-35) similar to that of Webb, thusly the modification and/or combination is proper, since the systems use similar principles of operation (9:45-10:15). Therefore, the line of argumentation that applicant presented in Remarks pages 8-9 is not available herein. Further, Curtis clearly does define regions by paths, which can be the outline of an object (e.g. its edges) – if applicant claims that the claim requires that such reference must **always** use edges, applicant is pointed to 7:47-51, where it is at least strongly suggested that

geometric objects consist of paths. Further, the system of Curtis allows the system to specify graphical objects and/or bounding rectangles (see how the system of Curtis can obtain bounding rectangles in 8:25-35). This would not change the principle of operation of the Politis reference. Rather, since regions can be defined by rectangles (7:35+) or paths, the system of Curtis would extend the system of Webb, and further allow it to use only the regions that overlap, not simply the bounding boxes, where the clipping region could then be defined as a path or a bounding box, as below.

Webb teaches a system wherein objects are defined by edges, and such objects are scan-converted on a line-by-line basis, where the trees in Figures 12-14 are generated to render them, and also has the active edges as noted above.

Therefore, it would be obvious to one of ordinary skill in the art to combine the instruction trees of Webb and Curtis with the graph structures of Katzenberger. The motivation would be to allow compiler-style optimization of rasterizing instructions and to enable the effective use of directed acyclic graphs as discussed by Politis (see Politis col. 10, lines 1-11 and 37-40) and also to derive the benefits of the DAG storage structures as shown by Katzenberger in those figures referenced above.

Further, Webb and Curtis are analogous art that are both directed to methods of optimizing and rendering graphic objects in various formats using DAGs – that is, tree formats. Curtis teaches additional information concerning the traversal of nodes in such a tree, and that such is done **after** scan-conversion, which means that the end results (particularly since the only nodes traversed would be those that were active and in the optimized tree of Politis anyway – see Figures 28 and 29, where the tree is optimized

anyway) would be optimized only traverse nodes that were active in the required manner anyway. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Webb in light of the teachings of Curtis to do the above limitations for the reasons enumerated above. Additionally and more importantly, the system of Politis does not take into account device-specific rendering issues (e.g. an object will be rendered in different ways with different devices), where the system of Curtis does, and thusly generates device-resolution specific regions that do not require scaling, and regions are freely transformable after combination, with many other limitations (2:30-57).

As to claims 2, 13, and 24, references Webb and Katzenberger teach all the limitations. Reference Katzenberger teaches a table for storing graphical instructions for acting on image components in Fig. 9, thus meeting the details recited in claim 2. See claim 1 for motivation and combination.

As to claims 3, 14, and 25, references Webb and Katzenberger teach all the limitations. This data is constructed as the tree is traversed and could easily be inserted into the table of Katzenberger shown in Fig. 9. See claim 1 for motivation and combination.

As to claims 5, 16, and 27, therefore, it would be obvious to one of ordinary skill in the art to combine the trees of Webb / Curtis with the graphic forms of Katzenberger. The motivation would be to have a tree structure to optimize as per claim 1 (see claim 1 references and col. 4, lines 1-10).

As to claims 6, 17, and 28, reference Curtis teaches the use of binary operators on the expression trees. See rejection to claim 1 for combination and motivation, as well as explanation, since the tree in for example Figures 6 and 7 are expression trees.

Allowable Subject Matter

Claims 7-11, 18-22, and 29-33 are allowed.

The following is a statement of reasons for the indication of allowable subject matter: the newly added limitation with ignoring dependent nodes if they are not active and the various other limitations, in the specific combination present therein, are not in the prior art singly, and it appears to examiner that an obviousness rejection involving 4-5 references against the independent claims would at best unwieldy and would lack motivation for such a combination in any case. It is pointed out for purposes of clarifying the record that the examiner feels that the strongest obviousness rejection that **could** be made would be Fraser in view of Curtis and probably Katzenberger and/or Webb. Again, however, such a combination would appear to lack a plausible motivation.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric Woods whose telephone number is 571-272-7775. The examiner can normally be reached on M-F 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Michael Razavi can be reached on 571-272-7664. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Eric Woods

Jeffrey A. Brier
PRIMARY EXAMINER

December 15, 2005